63-3-3

TM-1042/201/00

402 643

TECHNICAL MEMORANDUM

(TM Series)

This document was produced by SDC in performance of U. S. Government Contracts

SYSTEM DEVELOPMENT CORPORATION
2500 COLORADO AVE.
SANTA MONICA
CALIFORNIA

Duality and Network Flow

by

W. Karush

March 15, 1963

DUALITY AND NETWORK FLOW

by

W. Karush

1. INTRODUCTION

The principle of duality in linear programming is an important tool in the theory and application of mathematical programming. One of its significant uses is in the construction of computational procedures, so-called "primal-dual" algorithms, for the determination of optimal solutions of linear programming (LP) problems. Such an algorithm can be described for the general LP problem, but in this general case it is not competitive with the more standard procedures of solution, such as the simplex method. There are important classes of LP problems, however, which exhibit structures that allow primal-dual procedures with genuine computational advantages; these are problems that relate to optimal flows in networks, and it is this type of problem that we deal with here.

The present paper is an expository one whose purpose is to describe the notions of duality and network flow and to show how these are used in the construction of algorithms for certain classes of LP problems. The problems include the transportation problem, network flow at minimum cost, and scheduling of activities at minimum cost in a PERT-like, or critical-path, system of project organization. The techniques for the solution of these problems have appeared in the literature; the aim here is to provide a unified treatment of the underlying ideas without entering into a complete description of mathematical and computational details. References are provided for the reader who wishes to

pursue these details; particularly appropriate is the recent reference (1) (see list of references at end of paper) which contains full treatments and proofs of the algorithms discussed in this paper.

2. THE DUAL PROBLEM AND COMPLEMENTARY SLACKNESS

The algorithms to be discussed in this paper make use of the property of "complementary slackness," which relates the restraints of one LP problem with those of its dual problem. Let us begin by defining the dual problem. We remark at this point that the practice will be frequently followed of using a single letter to denote a point or vector whose components are denoted by subscripting that letter; e.g., $x = (x_1, x_2, \ldots, x_n)$.

Consider the LP problem (the "primal problem") in variables $x_1, x_2, \ldots, x_n$ of maximizing the objective function

$$\sum_{j=1}^{n} c_j x_j \tag{2.1}$$

subject to the m restraints

$$\sum_{j=1}^{n} a_{ij} x_j \leq b_i, \qquad i=1,2,\ldots,m\ , \tag{2.2}$$

and the restraints

$$x_j \geq 0, \qquad j=1,2,\ldots,n\ . \tag{2.3}$$

A point x satisfying the m+n restraints (2.2) and (2.3) is said to be a feasible point, or feasible solution, of the LP problem; a feasible point which maximizes

(2.1) is said to be an optimal point, or optimal solution. The purposes of this paper will be served best by avoiding pathological mathematical cases, so we shall assume for this problem, and any other considered in this paper, that an optimal point exists. (This means that there is at least one feasible point and that (2.1) is bounded from above in the set of all feasible points.)

The dual problem to the above is another LP problem given as follows: minimize

$$\sum_i u_i b_i \tag{2.4}$$

subject to

$$u_i \geq 0, \qquad i=1,2,\ldots,m\ , \tag{2.5}$$

$$\sum_i u_i a_{ij} \geq c_j, \qquad j=1,2,\ldots,n\ . \tag{2.6}$$

The variables in this problem are $u = (u_1, u_2, \ldots, u_m)$; notice that the number of components of u is the same as the number of restraints in (2.2) and that the number of restraints in (2.6) is the same as the number of components of x. (The components of u may be thought of as multipliers of the restraints (2.2); each column of the matrix of coefficients (a_{ij}) combined with these multipliers leads to one of the restraints (2.6) of the dual problem.) The restraints of the primal and dual problems (2.2), (2.3) and (2.5), (2.6), are written in the orders shown to display an intended correspondence between restraints and variables of the two problems; namely, the $i^{\underline{th}}$ component of u goes with the $i^{\underline{th}}$

inequality of (2.2) (or the $i^{\underline{th}}$ row of matrix (a_{ij})), and the $j^{\underline{th}}$ component of x goes with the $j^{\underline{th}}$ inequality of (2.6) (or the $j^{\underline{th}}$ column of (a_{ij})). Explicitly,

$$\begin{array}{ll} \Sigma a_{ij}x_j \leq b_i, & u_i \geq 0 \\ x_j \geq 0, & \underset{i}{\Sigma} u_i a_{ij} \geq c_j . \end{array} \tag{2.7}$$

The formal relationship between the two problems is also conveniently exhibited by a tabular array:

$$\begin{array}{c|cccc|c} & x_1 & x_2 & \cdots & x_n & (\geq 0) \\ \hline u_1 & a_{11} & a_{12} & \cdots & a_{1n} & b_1 \\ u_2 & a_{21} & a_{22} & \cdots & a_{2n} & b_2 \\ \vdots & \vdots & \vdots & & \vdots & \leq \ \vdots \\ u_m & a_{m1} & a_{m2} & \cdots & a_{mn} & b_m \\ \hline & & & \geq & & \\ (\geq 0) & c_1 & c_2 & \cdots & c_m & \end{array}$$

The u_i serve to index the (row) restraints on the x_j, and the x_j index the (column) restraints on the u_i.

For primal problems involving minimization of the objective function, an equivalent maximization problem is readily obtained by changing the signs of the c_j; the dual can then be formed according to the rules above. Similarly, any inequality of the ($\geq$) type in (2.2) can be changed to one of the ($\leq$) type by multiplying through by -1. If these devices are applied to the above dual

problem regarded as a primal problem, then the result comes out to be equivalent to the original primal problem; thus, the dual of the dual of a problem is the problem itself.

Complementary slackness has to do with implications relating associated pairs of restraints in (2.7) where one of the restraints is satisfied as a strict inequality. More precisely, let u be given; then x will be said to satisfy the condition of complementary slackness with respect to u (or, more briefly, to be complementary to u) in case for each of the m+n relations of (2.7) in u that is a strict inequality, the corresponding relation in x is a strict equality. That is, x is complementary to u in case

$$u_i > 0 \text{ implies } \sum_j a_{ij}x_j = b_i, \text{ and}$$
$$\sum_i u_i a_{ij} > c_j \text{ implies } x_j = 0. \tag{2.9}$$

Similarly, u is complementary to x in case

$$x_j > 0 \text{ implies } \sum_i u_i a_{ij} = c_j, \text{ and}$$
$$\sum_j a_{ij}x_j < b_i \text{ implies } u_i = 0. \tag{2.10}$$

The reason for the terminology "complementary slackness" can be explained if we interpret "slackness" to mean "strict inequality." Suppose x and u are feasible for their respective problems, and let the m+n pairs of relations in (2.7) be indexed successively by k=1,2,...,m+n. Let K be the set of index values for

which the u-restraints are slack (strict inequalities), and L for which the x-restraints are slack. Then the fact that x has the property of complementary slackness relative to u, as defined by (2.9), is equivalent to the fact that L is a subset of the complement of K (in the set of all index values). Notice that perfect complementary is not required, i.e., L need not equal the complement of K, so that there may be index values k for which equality holds for both variables. Sometimes the term "weak" complementary slackness is used for the property we have defined, and "strict" complementary slackness for the case when L equals the complement of K (when the statements (2.9) would be read as "if and only if" instead of "implies").

The discussion of the motivation of terminology just given supposed the feasibility of x and of u; however, in the definition of complementary slackness no such requirement was actually imposed on either x or u and, in fact, we shall want to make use of this freedom later on. However, we shall limit the use of the relationship to cases when one of the points is feasible, but not necessarily both; more specifically, we shall consider only situations of x being complementary to u when u is a feasible point. Under this circumstance, the implications (2.10) follow from the implications (2.9); thus, if x is complementary to a feasible u, then u is complementary to x (without necessarily being feasible).

We come to the connection between complementary slackness and optimality, a connection which lies at the core of the algorithms we shall be discussing. It is this: if u is a feasible point of the dual problem, and if x is both complementary to u and a feasible point of the primal problem, then x and u are optimal points of their respective problems. The proof of this important fact

is not difficult; it may be outlined as follows. First we readily establish that for any pair of feasible points x and u (not necessarily complementary), the respective objective functions satisfy the inequality

$$\sum_j c_j x_j \leq \sum_i u_i b_i . \tag{2.11}$$

This carries the consequence that if any pair of feasible points satisfy (2.11) as an equality, then they are necessarily optimal points. Then we argue that complementary slackness of feasible points does indeed imply equality in (2.11).

Let us consider next the modifications needed in the preceding discussion when the primal LP problem, instead of being phrased in the standard form of maximizing (2.1) subject to (2.2) and (2.3), is given as maximizing (2.1) subject to

$$\sum_j a_{ij} x_j = b_i, \qquad i=1,2,\ldots,m \;, \tag{2.2'}$$

$$x_j \geq 0, \qquad j=1,2,\ldots,n \;. \tag{2.3'}$$

This is the format generally used in the simplex algorithm. The dual problem in this case takes the form of minimizing (2.4) subject to

$$u_i \text{ unrestricted}, \qquad i=1,2,\ldots,m \;, \tag{2.5'}$$

$$\sum_i u_i a_{ij} \geq c_j, \qquad j=1,2,\ldots,n \;. \tag{2.6'}$$

An important feature to be noticed is that equality restraints (2.2') on x

correspond to no conditions (unrestricted) on associated dual variables (2.5'). The new dual form can be derived from the rule for dualizing (2.1), (2.2), (2.3); to do this, we write each equality of (2.2') as a pair of inequalities, $\leq$ and $\geq$, then reverse the second by multiplying by -1. The dual is then formed in 2m non-negative variables, say v_i, w_i; but these variables only occur in the combination $v_i - w_i$, and introducing a variable u_i for this difference leads to (2.5'), (2.6').

We may consider the general case of mixed restraints in which some of the relations in (2.7) are inequalities and others are equations, and where some of the variables in (2.3) are unrestricted. This may be expressed as maximizing (2.1) subject to

$$\sum_j a_{ij}x_j = b_i, \qquad i=1,2,\ldots,p, \tag{2.2a}$$

$$\sum_j a_{ij}x_j \leq b_i, \qquad i=p+1,p+2,\ldots,m, \tag{2.2b}$$

and

$$x_j \geq 0, \qquad j=1,2,\ldots,q, \tag{2.3a}$$

$$x_j \text{ unrestricted}, \qquad j=q+1,q+2,\ldots,n. \tag{2.3b}$$

(Here, sums over j are carried out for the full range $j=1,2,\ldots,n$.) The dual problem becomes that of minimizing (2.4) subject to

$$u_i \text{ unrestricted}, \qquad i=1,2,\ldots,p, \tag{2.5a}$$

$$u_i \geq 0, \qquad i=p+1,p+2,\ldots,m\ , \qquad (2.5b)$$

$$\sum_i u_i a_{ij} \geq c_j, \qquad j=1,2,\ldots,q\ , \qquad (2.6a)$$

$$\sum_i u_i a_{ij} = c_j, \qquad j=q+1,q+2,\ldots,n\ . \qquad (2.6b)$$

(Sums are carried out for the full range of i.) In this general case we still take complementary slackness to be defined by (2.9) (and (2.10)) with the proviso that only those values of the unsummed subscript in (2.9) are considered corresponding to inequalities in the LP problem; i.e., in (2.9) the range of i and j are $p+1,\ldots,m$ and $1,2,\ldots,q$, in the first and second set of implications, respectively. For example, in the case of (2.2'), (2.3') and (2.5'), (2.6') (where $p=m$ and $q=n$), we retain only the second set of implications in (2.9) (and the first in (2.10)). This modification of the definition in the general case does not invalidate the relation between complementary slackness and optimality; we still have the fact that if u is feasible and x is complementary to u, then feasibility of x implies optimality of x (and of u).

It may be noted that the converse of the last statement can also be proved, i.e., optimality of x and u implies their complementarity. A final feature will be useful in the coming discussion; this is that algorithms yielding an optimal solution of an LP problem (e.g., the simplex method) will at the same time yield an optimal solution to the dual problem.

3. A GENERAL PRIMAL-DUAL METHOD

In order to communicate the essential ideas of this section in the simplest

way and without an undue amount of notation, we shall treat the primal LP problem in the standard form of maximizing (2.1) subject to (2.2') and (2.3'). The transfer to other forms causes no difficulty.

The discussion in the last section shows that the question of maximizing (2.1) subject to (2.2') and (2.3') can be transformed to an equivalent problem of finding two points x and u which satisfy (2.2'), (2.3') and (2.6'), respectively, together with the condition of complementary slackness, namely,

$$\sum_i u_i a_{ij} > c_j \text{ implies } x_j = 0. \tag{3.1}$$

This replaces a maximization problem in n variables x_j by a problem of solving a system of inequalities and equations in m+n variables u_i, x_j. This would appear to be an advantage, on the face of it; the difficulty, of course, lies in the form of the requirement (3.1), which is not a direct linear restraint like the others but a condition on the way these direct restraints are to be satisfied. Nevertheless, the alternative formulation suggests a procedure for searching for an optimal solution which we now wish to describe.

Suppose that a particular feasible point u^* is at hand, obtained by, say, inspection of (2.6') or by any other means. Let us confine attention to those points x which are complementary to u^* and attempt to solve the system of inequalities (2.2'), (2.3') within this set of points; i.e., in addition to imposing (2.2') and (2.3') we suppress those components x_k (to 0) for which the $k^{\underline{th}}$ relation (2.6') is satisfied as a strict inequality by the given dual point u^*. For the time being, let the index β range over the subscript values

corresponding to suppressed components of x and α over the remaining subscripts. Then we seek a solution of

$$\sum_j a_{ij}x_j = b_i, \qquad i=1,2,\dots,m\,,$$
$$x_\alpha \geq 0,\ x_\beta = 0. \tag{3.2}$$

Unfortunately, it is not to be expected in general that (3.2) has a solution, for if it did, then the solution would be an optimal point for the primal problem (by the relationship between complementary slackness and optimality), and we would have had the good fortune of having picked out an optimal dual point u^* to begin with.

The idea that is introduced to attempt to circumvent this difficulty is to abandon the requirement that the quantities b_i be constants and, instead, treat them as parameters, or variables. Thus, instead of (3.2) we consider the system

$$\sum_j a_{ij}x_j = y_i$$
$$x_\alpha \geq 0,\ x_\beta = 0 \tag{3.3}$$

in variables x_j, y_i. The usefulness of (3.3) is based in part upon the fact that the complementary slackness condition (3.1) does not depend upon the b_i; hence, any solution (x,y) of (3.3) will provide a point x, while not necessarily feasible for the primal problem, is complementary to the feasible u^*. Even more can be said: a solution (x,y) of (3.3) provides an optimal point x for that modified primal problem in which the b_i of (2.2') are replaced by the specified y_i.

But now we face another difficulty. Although there is no difficulty in finding solutions (x,y) of (3.3) (e.g., take an arbitrary x satisfying the second line of (3.3) and determine y by the first line), we have seen that we cannot expect to hit upon a solution with $y=b$; having found a point x^* complementary to u^* through some solution (x^*,y^*) of (3.3) how shall we go about "improving" x^* to move closer to the desired optimal point? In order to answer this question, we first introduce a measure of the closeness of a complementary point to an optimal point. Let us adjoin the conditions

$$y_i \leq b_i, \qquad i=1,2,\ldots,m \tag{3.4}$$

to (3.3). Then it is plausible to take

$$\sum_i y_i \tag{3.5}$$

as an indicator of closeness to an optimal point for any x corresponding to a solution (x,y) of (3.3), (3.4), for this quantity never exceeds the fixed value $\sum_i b_i$, and it can attain this value only when x is optimal. The way to improve x, therefore, is to increase (3.5).

The essential idea of a primal-dual procedure may now be sketched out. With the given primal problem and a feasible point u^* of its dual problem, associate a so-called <u>restricted primal problem</u>, namely, to maximize (3.5) subject to (3.3) and (3.4). Solve the restricted primal for the given u^*; this yields a point x^* which is a closest approximation to an optimal point among the points complementary to u^*. Laying aside for a moment the objection that this itself

requires finding an optimal solution of an LP problem (so why not solve the original problem directly?), we proceed to the matter of improving the approximation. This is to be done by modifying the starting feasible dual point u^* to a new one u^{**} which will result in an improved x^{**} when the restricted primal is solved which is prescribed by u^{**}. This modification is achieved by making use of an optimal solution of the dual restricted problem, i.e., the dual of the restricted primal problem; recall that such a solution is automatically at hand when the restricted primal (associated with u^*) is solved. Let us discuss in further detail how this is done. (The reader who is not concerned with elaboration of this point may omit the following paragraph.)

For the formulation of the dual restricted problem, it is helpful to state the restricted primal masking out the suppressed components of x:

$$\text{maximize } \sum_i y_i$$

subject to

$$\begin{aligned} \sum_\alpha a_{i\alpha} x_\alpha - y_i &= 0 \\ y_i &\leq b_i \end{aligned} \tag{3.6}$$

with

$$x_\alpha \geq 0, \quad y_i \text{ unrestricted.}$$

Dualizing this problem with multiplier variables μ_i, σ_i according to the rule

in the last section, leads to:

$$\text{minimize } \sum_i \sigma_i b_i$$

subject to

$$\mu_i \text{ unrestricted, } \sigma_i \geq 0,$$

and

$$\begin{aligned} &\sum_i \mu_i a_{i\alpha} \geq 0 \\ &-\mu_i + \sigma_i = 1 \end{aligned} \tag{3.7}$$

Suppose now that we have solved the restricted primal defined by u^* and have at hand an optimal solution (x^*, y^*) of the restricted primal and one (μ^*, σ^*) of this dual problem. What is done is to choose as a new dual point u^{**} a linear combination

$$u^{**} = u^* + \theta\mu^* \ , \tag{3.8}$$

where θ is an approximately selected positive number. To see how θ is selected, observe that the feasibility condition (2.6') on u^{**} requires that

$$(\sum_i u_i^* a_{ij}) + \theta(\sum_i \mu_i^* a_{ij}) \geq c_j, \quad j=1,2,\ldots,n \ . \tag{3.9}$$

By the inequalities (2.6') and (3.7) this holds for all $\theta \geq 0$ when $j=\alpha$ (index

values of the unsuppressed components). For the remaining index values β, (2.6') is a strict inequality; hence, no matter what the sign of $\sum_i \mu_i^* a_{i\beta}$, it is possible to choose $\theta > 0$ so that (3.9) also holds for $j=\beta$. Thus, (3.8) does provide a feasible dual point with which to regenerate the computation; in addition, it can be argued that x^* is complementary to u^{**}, as well as to u^*.

This completes the description of the general step in an iterative primal-dual algorithm. When a new step of the iteration is begun with the initial dual point u^{**}, the most recent (x^*,y^*) serves as an initial point for the process used in solving the new restricted primal; this ensures that the (x^{**},y^{**}) obtained as the optimal point of that restricted primal will be an improvement over the last point (x^*,y^*), i.e., we will have $\Sigma y_i^{**} \geq \Sigma y_i^*$. The iteration will terminate in a finite number of steps, when the optimal value of the restricted primal reaches $\sum_i b_i$, i.e., when $y_i=b_i$ for all i.

Let us return to the point put aside earlier, namely, the need of solving an LP problem, a restricted primal, in each cycle of the primal-dual algorithm. If this restricted problem is of such a nature that its solution calls for the application of a general LP algorithm , such as the simplex method, then the proposed method had best be discarded in favor of the direct solution of the primal problem by the general method itself. However, if the restricted primal has such a structure that a simplified procedure may be used for its solution which is not applicable to the primal, then the proposed method may be competitive with or preferable to a general method which does not specifically take advantage of the special structure of the problem. This is indeed the case for problems where the solution of the restricted primal comes down to the determination of

the maximal flow in a network. The next section, then, will discuss flows in networks; after that we may proceed to LP problems that can be efficiently solved by a primal-dual method combined with a network flow algorithm.

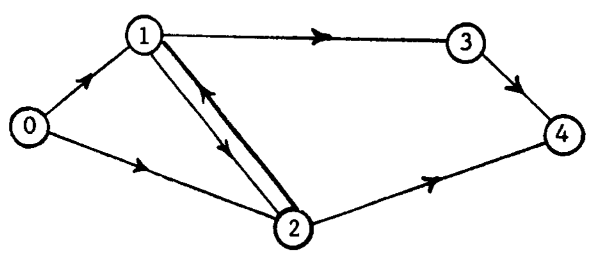
A final remark is in order before ending this section. In the foregoing, we chose to parameterize the restraint constants b_i and selected $\sum_i y_i$ as a means of indicating the closeness of a complementary point to an optimal point of the primal problem. This was convenient for the form of LP problem which was taken as a point of departure for the present exposition. However, it may be desirable to use some other indicator in dealing with a type of problem having a particular structure, or to parameterize the constants in some other manner which is intrinsic to the particular problem at hand. Such alternatives, based on the general primal-dual approach outlined here, offer possibilities to be explored in attempting to discover solution algorithms that make use of the special structure of a mathematical programming problem.

4. FLOWS IN NETWORKS

In this section we present a brief description of some of the basic notions of networks and flows in networks, with the treatment being intended to serve only the immediate purposes of this paper; a more complete discussion will be found in reference 1.

A (directed) network W is a finite collection of points, or nodes, together with a selected subset of the set of all ordered pairs of nodes, or (directed) branches. (We suppose that each node belongs to at least one ordered pair in W.) Let N+1 be the number of nodes of W, and designate the nodes as 0,1,...,N. For

branch (r,s) belonging to W, we require that r $\neq$ s, excluding, thereby, branch loops (r,r) (but not excluding loops of two or more branches). The nodes of W may be represented by points (or small circles) in the plane and the branches by directed line segments or arcs. (We have assumed at most one branch from an initial point r to a terminal point s, although there is no real difficulty in handling networks with several branches joining the same nodes in the same order; indeed we shall be considering such a situation in a later section.) A source (sink) of W is a node which appears only at the initial (terminal) point of branches of W. For simplicity, we assume that W possesses a unique source and a unique sink, and denote these by 0 and N, respectively; this is no genuine restriction, because a multiple-source and multiple-sink network can be easily modified to the simpler type by adding an artificial sink, and some additional branches. Any node other than the source or sink is termed an intermediate node.



By a (directed) chain in a network W from one node r to another node s is meant a sequence of branches which form a connected path from r to s when the branches are traversed along their given directions and where it is assumed that

the path contains no closed loops. By simply a path from r to s is meant the same except that no requirement is imposed that the branches be traversed in the given directions; thus, traversing a path from r to s may involve going contrary to the direction of some of its branches. For example, in the figure, (0,1), (1,3) comprise a chain from 0 to 3, while (0,2), (3,2) comprise a path from 0 to 3 which is not a chain; also, (0,2), (2,1) is a chain from 0 to 1 but (0,2), (1,2) is a path from 0 to 1. We shall be especially interested in paths from the source to the sink.

We turn now to the notion of a flow in a network W. The intuitive idea is expressed by a fluid flowing at a steady rate through a network of channels (branches) with the direction of flow specified in each channel: the fluid enters the network at a steady rate f, distributes itself among the channels as a steady stream along the given direction in each channel, and emerges from the network at the sink at the same steady rate f at which it entered. Each channel is taken to have a maximum capacity (possibly ∞) which specifies the greatest rate of flow it can accommodate. Conservation of flow is assumed at each node, which means that the net flow out of an intermediate node is 0 (flow emerging from r = flow entering r). The problem of maximum flow is to find those channel flows in W for which f has the largest possible value under the given capacity limitations.

This problem can be expressed mathematically as an LP problem. Consider a given network W with a capacity K_{rs} associated with each branch (r,s) of W. Introduce variables X_{rs}, f (where X_{rs} is to be interpreted as the flow along (r,s) and f as the net flow through W). Then a set of values of these variables

is feasible in case

$$\left.\begin{aligned} \sum_s X_{0s} &= f, && (r=0)\ , \\ \sum_s X_{rs} - \sum_t X_{tr} &= 0, && r\neq 0,N\ , \\ -\sum_t X_{tN} &= -f, && (r=N)\ , \end{aligned}\right\} \tag{4.1}$$

$$X_{rs} \leq K_{rs}\ , \tag{4.2}$$

$$X_{rs} \geq 0\ . \tag{4.3}$$

Here summations are taken over values of indices corresponding to branches of W, and in (4.2) and (4.3) the double subscript ranges over all branches of W. Equations (4.1) are balance-of-flow equations, the left side, in each case, being the net flow out of a node; the first and third apply to the source and sink, respectively, and the second to intermediate nodes. The maximal flow problem, then, is to

$$\text{maximize } f \tag{4.4}$$

with the variables X_{rs},f subject to (4.1), (4.2), (4.3). Notice that equations (4.1) are not independent; the last is obtained by adding all the preceding ones. The redundant restraint will be kept, however, for symmetry of notation.

Various network flow models which appear to be more general than the preceding one can be put in this simpler form. We have already mentioned the

case of multiple sources and sinks. Another case is that in which capacity limitations are imposed on the nodes, as well as on the branches; the device that may be used here is to split a node into two nodes and transform the node capacity to a branch capacity on the new branch joining the two nodes. A reason for the importance of the maximal flow problem is the variety of problem types that can be related to it.

The particular structure of the maximal flow problem, as contrasted with the general LP problem, permits the invention of an algorithm for its solution which is considerably more efficient than a general procedure for solving any LP problem, such as the simplex method. We refer to the labeling technique of Ford-Fulkerson. The underlying idea of that method is straightforward; it is an iterative procedure which at each stage begins with a given net flow f in the network, searches for some path from source to sink which can carry an incremental flow, when it finds such a flow-augmenting path modifies the given flow accordingly, and then repeats the process. A flow-augmenting path which is a chain carries an incremental amount ϵ along each of its branches in the forward direction; this is used to increase f to $f+\epsilon$ by adding ϵ to each branch variable on the path, and adding 0 to other branch variables. When the flow-augmenting path of amount ϵ is not a chain, i.e., traverses some branches in the reverse direction, then ϵ is to be subtracted from those branch variables corresponding to contrary traversal; this still achieves a net increase to $f+\epsilon$ in the total network.

The labeling technique of Ford-Fulkerson is the bookkeeping procedure used at each stage to search for a flow-augmenting path. It starts at the source 0

and proceeds to label subsequent nodes in a simple, systematic fashion; when any particular node is reached and labeled, this means that a path from source to that node is known which can carry flow. The labeling terminates in one of two ways: either the sink is eventually labeled ("breakthrough"), or the labeling process stops without reaching the sink ("nonbreakthrough"). In the first case the network flow is augmented, the labels wiped out, and a new labeling of nodes begun; in the second case, the network flow with which the labeling began is in fact the maximal flow. The algorithm will terminate in a finite number of stages, provided the capacities K_{rs} are integers (this includes the case of rational numbers; simply take the unit of flow as $1/q$, with q a common denominator of the capacities). From the practical, computational point of view this is no real restriction. The optimal values of the variables generated by the algorithm will themselves then be integral.

The problem dual to the maximal flow problem (4.1)-(4.4) involves dual node variables U_r, one for each of the equations (4.1), and dual branch variables V_{rs}, one for each of the inequalities (4.2). The dual problem is the following:

$$\text{minimize } \sum_{r,s} K_{rs} V_{rs}$$

subject to the conditions

$$U_r \text{ unrestricted}, \; V_{rs} \geq 0 \; , \tag{4.5}$$

and

$$\begin{aligned} U_r - U_s + V_{rs} &\geq 0 \; , \qquad (r,s) \text{ in } W, \\ - U_o + U_n &= 1 \; ; \end{aligned} \tag{4.6}$$

the last relation (equality) corresponds to the column of coefficients of the primal variable f in (4.1), and the one before it (inequality) to the column of coefficients of X_{rs} in (4.1) and (4.2). The labeling technique determines an optimal solution for this problem at the same time as it does for its primal; these optimal variables are integral (in fact, either 0 or 1).

As we mentioned in the previous discussion of the primal-dual method, the maximal flow problem will occur in applications as a restricted primal. In that context, in addition to the conditions (4.1)-(4.3), the primal variables will be subject to certain suppression conditions dictated by the requirement of complementary slackness. These conditions will call for holding designated branch flows X_{rs} at the minimum value 0 and others at their maximum value K_{rs}, and the maximal flow problem will need to be solved with these conditions adjoined. The maximal flow algorithm can accommodate such simple additional requirements with no difficulty at all; the flow-augmenting paths are simply limited to paths which do not contain any branch with a suppressed variable.

5. THE TRANSPORTATION PROBLEM

The first type of LP problem for which we shall describe a primal-dual algorithm is the well-known transportation problem. There are n points of destination, s=1,2,...,n, at each of which a specified demand is to be met for a certain commodity, and there are m points of origin, r=1,2,...,m, at each of which there is at hand a specified supply of that commodity. The demands are to be met by shipping from the origins to the destination; we assume that the total given quantity Q at all the origins is equal to the total demand at all the destinations. The cost c_{rs} of shipping one unit of the commodity from a origin

r to a destination s is known, and the cost of shipping any number of units is proportional to that number. The problem is to determine a program of shipments from origins to destinations which will minimize the total shipping cost.

Let X_{rs} denote the number of units shipped from origin r to destination s; then the problem is to

$$\text{minimize } \sum_{r,s} c_{rs} X_{rs} \tag{5.1}$$

subject to

$$\begin{aligned} &\sum_t X_{rt} = p_r , &\qquad r=1,2,\ldots,m , \\ &\sum_t X_{ts} = q_s , &\qquad s=1,2,\ldots,n , \end{aligned} \tag{5.2}$$

and

$$X_{rs} \geq 0 , \qquad \text{all } (r,s) .$$

Here p_r is the given supply at r, q_s the given demand at s, and we require that

$$\sum_r p_r = \sum_s q_s = Q .$$

It may be remarked that various apparently more general forms of the transportation problem can be cast in this standard form.

Let us interpret the primal-dual procedure described in Section 1 for this primal problem. Assigning the dual-origin variable U_r to the first equation of (5.2) and the dual-destination variable V_s to the second equation, we see that

the restraints of the dual problem can be expressed as follows:

$$U_r + V_s \leq c_{rs} , \qquad U_r, V_s \text{ unrestricted.} \tag{5.3}$$

(To see this from (2.2'), (2.3') and (2.6') in Section 2, express (5.1) as a maximization problem by using $-c_{rs}$ in place of c_{rs}, apply the rule in Section 1 for writing the dual problem, then replace the dual variables by their negatives.)

To obtain the restricted primal problem we replace the constants p_r, q_s in (5.2) by variables Y_{or}, Z_{sN}; the constraints of that problem then have the form

$$\begin{aligned} &\sum_t X_{rt} - Y_{or} = 0 \\ &Z_{sN} - \sum_t X_{ts} = 0 \end{aligned} \tag{5.4}$$

with

$$0 \leq Y_{or} \leq p_r , \quad 0 \leq Z_{sN} \leq q_s , \quad X_{rs} \geq 0 . \tag{5.5}$$

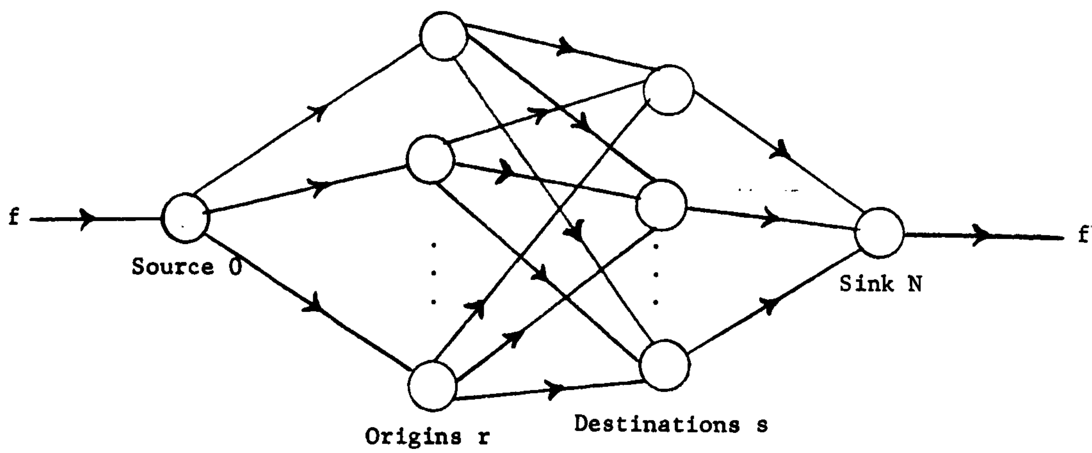
The function to be maximized is $\sum_r Y_{or} + \sum_s Z_{sN}$; let us modify this slightly. Introduce the variable f and set

$$\sum_r Y_{or} = f . \tag{5.6}$$

Summing the first equation (5.4) over r and the second over s, we see that

$$- \Sigma Z_{sN} = -f . \tag{5.7}$$

Since the sum to be maximized equals 2f, we may state the restricted primal as that of maximizing f instead, subject to (5.4)-(5.7) (delaying consideration of suppressed variables for the moment). Now this may be recognized as a maximal-flow problem in a network W in the following way. Take each origin and each destination as a node, and take each ordered pair (origin, destination) as a branch; in addition, introduce one source 0 and one sink $N=m+n+1$ and include among the branches all pairs (source, origin) and all pairs (destination, sink). Assign the capacity p_r, ∞, q_s, respectively, to the branches (0, origin r), (origin r, destination s), (destination s, N); the restricted primal problem is that of maximizing flow in this network, where Y_{or}, X_{rs}, Z_{sN} are the flows in these branches, respectively (equations (5.6), (5.4), (5.7) are the conservation equations at the source, and origins and destinations, and sink, respectively, for a net flow of f in W, and (5.5) are the capacity conditions).



In the primal-dual algorithm, the restricted problem is associated with a dual point which specifies the suppression of certain variables. Let U, V be a dual feasible point, i.e., let it satisfy (5.3). Complementary slackness of branch flows X_{rs} requires

$$U_r + V_s < c_{rs} \text{ implies } X_{rs} = 0 .$$

Thus, the maximal flow is to be found with those branch flows held at 0 for which the hypothesis of this implication is valid; this problem is solved efficiently by the labeling algorithm outlined in the last section.

The procedure for determining the optimal solution of the transportation problem, then, goes as follows. Start with some feasible dual point U, V (for example, take all U_r, V_s equal to 0 when $c_{rs} \geq 0$). Solve the associated restricted primal (maximal flow) problem by the labeling technique, obtaining a maximal flow f, optimal values Y_{or}, X_{rs}, Z_{sN}, and a set of optimal values u_r, v_s for the dual node variables corresponding to the nodes which are origins and destinations. Modify U, V to another feasible dual point $U^* = U + \theta u$, $V^* = V + \theta v$, for an appropriate value of θ. Now repeat the entire process beginning with U^*, V^*; the procedure for finding the maximal flow in this step can begin with the initial flow values Y, X, Z equal to the optimal flow values found at the end of the preceding step. The steps are repeated until the optimal f of the restricted primal attains the value $Q = \Sigma p_r$; at this point the optimal solution of the restricted primal is the optimal solution of the original primal problem (because it is feasible for the primal as well as complementary to a feasible dual point).

With regard to the computational efficiency of this network flow method, the following may be said. Although the general simplex method for solving LP problems can be simplified when account is taken of the special structure of the transportation problem, computational experience seems to indicate that the network flow method is to be preferred.

6. FLOW THROUGH NETWORK AT MINIMUM COST

We now consider a generalization of the transportation problem which involves a more complicated network than the one occurring in that problem. Let W be a given network and consider any flow through the network, as described in Section 4. Suppose the cost of accommodating the branch flow X_{rs} through the branch (r,s) is given by $c_{rs}X_{rs}$; let the costs of the various branch flows be independent, so that the total cost of the program X_{rs} is given by

$$\sum_{r,s} c_{rs}X_{rs} \ . \tag{6.1}$$

Let Q be a given value of the net flow, between 0 and the maximal flow, which is to be carried through W. In general, there will be many programs of branch flows which will realize the net flow Q; the problem we wish to consider is that of determining an optimal such program, i.e., one that minimizes the total cost (6.1).

The mathematical formulation of the problem is obtained by replacing the variable f in (4.1) by the constant Q and requiring among all programs X_{rs}, which are feasible in terms of the resulting conditions (4.1), (4.2), (4.3), the one that minimizes (6.1). The dual problem can be put in the following form,

similar to (4.5), (4.6). Let U_r, V_{rs} be dual node and branch variables respectively; then a set of values of these variables is feasible in case

$$U_r - U_s - V_{rs} \leq c_{rs} \ , \tag{6.2}$$

$$U_r \text{ unrestricted, } V_{rs} \geq 0 \ , \tag{6.3}$$

for every branch (r,s). The objective function is

$$Q(U_o - U_N) - \sum_{r,s} K_{rs} V_{rs} \ ,$$

which is to be maximized. The complementary slackness conditions on flow variables X_{rs} may be expressed as follows:

$$V_{rs} > 0 \text{ implies } X_{rs} = K_{rs} \text{ (saturated)} \tag{6.4}$$

$$U_r - U_s - V_{rs} < c_{rs} \text{ and } V_{rs} = 0 \text{ implies } X_{rs} = 0 \text{ (inactive).} \tag{6.5}$$

As noted, a branch is saturated if its flow is fixed at the maximum value for that branch, and inactive if fixed at 0; a branch is active in case it is not inactive (it may be saturated).

To derive the restricted problem we begin by following the procedure of Section 3; the variable f is restored in place of Q, giving restraints which read precisely as (4.1), (4.2), (4.3); in addition, we adjoin the requirement

$$f \leq Q \ .$$

This set of conditions describes a flow through W of net value f not exceeding Q; hence, the appropriate function to maximize in the restricted primal is f itself, since increasing f will force the program X_{rs} for the restricted problem as close as possible toward the desired net flow Q. In this way, the restricted primal becomes a maximal flow problem.

The primal-dual algorithm, then, proceeds as follows. Pick a feasible dual point U, V (say, identically 0 at the first stage, if all $c_{rs} \geq 0$). Determine the saturated and inactive branches by conditions (6.4), (6.5). Solve the maximal flow problem for the network by the algorithm of Section 4, holding saturated and inactive branches at their preassigned values. If the flow f reaches the value Q in the course of the maximal flow algorithm, the minimum cost solution has been reached and the computation is terminated. If the maximal flow f^* is less than Q, use the values of the dual variables for the restricted primal obtained along with f^* to modify the starting U, V to a new feasible dual point; repeat the process with this feasible dual point, defining new saturated and inactive branches, etc. Iterate until the flow Q is finally reached.

This computational procedure is essentially a parametric method wherein the net flow in W is treated as the parameter. In effect, the procedure generates the minimum cost of carrying the net flow Q as a continuous function of Q (as well as the program realizing that cost), for Q ranging over the interval from 0 to the maximal flow in the network; the graph of the minimum cost against Q is increasing, piecewise linear, and concave (marginally decreasing).

An extension of the minimum-cost flow problem has appeared in the literature

(reference 3) which allows network flows with gain factors. In such a network, the branch flow X_{rs} leaving node r along (r,s) is multiplied by a factor g_{rs} before entering node s, so that the flow actually entering s from (r,s) is $g_{rs}X_{rs}$. This breeding or loss effect may be interpreted in various ways depending upon the application--spoilage in storage, interest rate, etc. Conservation of flow at each node is assumed (flow entering = flow leaving). When the network is normalized so that there is a single source and no sink (which can always be accomplished) the problem is expressed mathematically as follows. Minimize (6.1) subject to

$$\sum_s X_{os} = Q$$

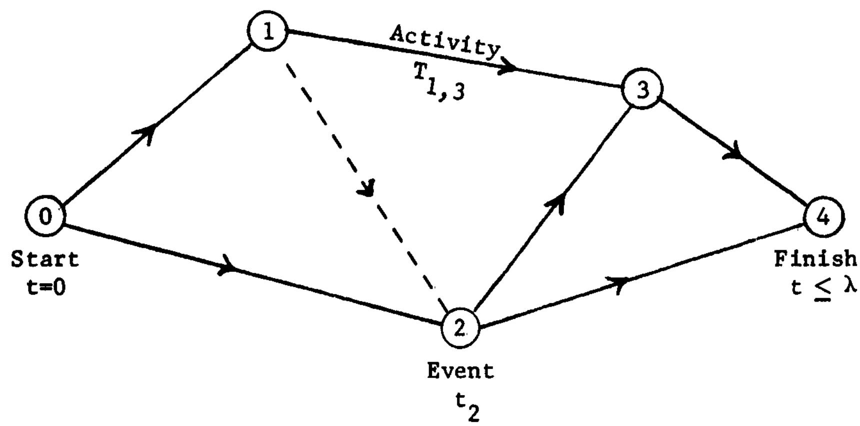
$$\sum_s X_{rs} - \sum_t g_{tr}X_{tr} = 0$$

and $0 \leq X_{rs} \leq K_{rs}$. The optimal program can be obtained by a primal-dual algorithm similar to the preceding, but one that is complicated by having to determine maximal flow in a network with gain factors as the restricted primal, rather than maximal flow in an ordinary network. As it turns out, this can be managed by a generalization of the labeling technique, but at a fairly high price in computational simplicity.

7. PROJECT SCHEDULING AT MINIMUM COST

The problem to be discussed here deals with a project composed of a number of subprojects, or activities, in which the activities are partially ordered in time, i.e., in which certain activities need to be completed before others may

begin. Each activity is to be assigned a duration time whose value must lie between a least value (crash time) and a greatest value (normal time) for that activity; the cost of accomplishing a given activity is a function of the assigned duration time and decreases linearly from a greatest cost for the crash time to a least cost for the normal time. Suppose the entire project, beginning at time 0, is to be completed by a given point in time λ; the objective is to program the activities (assign duration intervals to the activities and schedule these intervals over time) so that this deadline is met at minimum project cost, this being the sum of the costs of the individual activities. There will be a least time λ_o in which the project can be completed; the problem we wish to treat is, more generally, the parametric one of determining the minimum cost (and related optimal program) as a function of λ, for $\lambda \geq \lambda_o$.



It is convenient to represent the precedence relations among activities by means of a network W. Activities are described by the (directed) branches (r,s) of the network W; the fact that the terminal node of one activity is the initial node of another expresses that the first activity must be completed before the second can begin. Thus, the (immediate) predecessors of any activity are given by all the activities (branches) that lead into it and its (immediate) successors by all those that lead out of it. By the use of dummy activities it is possible to express all precedence relations by means of a network in which no two activities share common initial and terminal nodes. For example, the above figure is intended to illustrate a project in which activity (1,3) must await the completion of (0,1), while (2,3) must await the completion of both (0,1) and (0,2); this requirement is met by introducing a dummy activity (1,2), shown dashed. Such dummy activities are assigned duration times and costs of zero. (See reference 4 for this and other devices useful in constructing a network representation of a realistic project.)

The nodes of W are known as "events," or "milestones"; let them be indexed by 0,1,2,...,N. Each event r is to occur at some point in time, say, t_r; there is a start event 0, assigned the time $t_o = 0$, and a finish event N, whose assigned time must satisfy $t_N \leq \lambda$, where λ, as mentioned, is the required completion time of the project. Because the network contains no closed (directed) chains, we may suppose the events have been indexed so that (r,s) is an activity only if $r < s$; also, for every node r there is a chain from 0 to r, and a chain from r to N. Now if T_{rs} denotes the duration time of the activity (r,s), then the precedence relations are met by requiring that $T_{rs} \leq t_s - t_r$,

i.e., the duration time of an activity cannot exceed the time interval between its initial event and its terminal event (note that the duration time may be strictly less than this interval).

It remains only to describe explicitly the cost function to complete the mathematical elements of the problem. Let L_{rs} and M_{rs} denote the least duration time and maximum duration time, respectively, for the activity (r,s); we must have $L_{rs} \leq T_{rs} \leq M_{rs}$. Let $D_{rs} - C_{rs}T_{rs}$, with $D_{rs} \geq 0$, $C_{rs} \geq 0$, give the cost of activity (r,s); then the cost of the project is the sum of these costs over all activities, and the objective is to minimize this sum, or, what is the same thing, to maximize

$$\sum_{r,s} C_{rs}T_{rs} \quad . \tag{7.1}$$

The mathematical problem comes down to this, then. Given the network W of events and activities, a feasible schedule is a set of values for the event and activity variables t_r and T_{rs}, respectively, which satisfies the following restraints for every (r,s) in W:

$$\begin{aligned} T_{rs} + t_r - t_s &\leq 0, \\ t_N &\leq \lambda, \\ T_{rs} &\leq M_{rs}, \\ - T_{rs} &\leq - L_{rs}. \end{aligned} \tag{7.2}$$

(The non-negativity of the variables need not be assumed explicitly; this is implied by $t_o = 0$ and the first and fourth inequalities.) We wish to determine a feasible schedule which maximizes the expression (7.1); this maximum is a

function $F(\lambda)$ of λ, and the problem is, further, to determine its value for each λ, as well as an optimal schedule for each λ. It may be pointed out here that $F(\lambda)$ is a decreasing, piecewise linear, convex function over its domain of definition $\lambda \geq \lambda_o$, where λ_o, as mentioned, is the minimum λ for which a feasible schedule exists (also $F(\lambda)$ is constant for sufficiently large λ). Also, notice that the restraints (7.2) may be viewed as follows, in light of the aim of maximizing (7.1): let occurrence times t_r of events satisfy the second equation (7.2) and the first equation (7.2) with T_{rs} replaced by L_{rs}; then determine the duration times as the smaller of M_{rs} and $t_s - t_r$, i.e.,

$$T_{rs} = \min[M_{rs}, t_s - t_r] . \tag{7.3}$$

(Notice that if $M_{rs} < t_s - t_r$, then activity (r,s) has slack, and the duration interval M_{rs} may be positioned anywhere between t_r and t_s.)

Following the method of references 1 and 2, we take as the primal problem not the above formulation but its dual, instead. The dual is stated in terms of non-negative variables X_{rs}, f, Y_{rs}, Z_{rs}, corresponding to each of the restraints (7.2), and are subject to

$$X_{rs} + Y_{rs} - Z_{rs} = C_{rs}, \quad (r,s) \text{ in } W \tag{7.3}$$

$$\sum_s X_{rs} - \sum_t X_{tr} = \begin{cases} 0, & r \neq 0, N, \\ -f, & r = N. \end{cases} \tag{7.4}$$

Under these constraints, we are to minimize

$$\lambda f + \sum_{r,s} (M_{rs} Y_{rs} - L_{rs} Z_{rs}). \tag{7.5}$$

Without going into details, we may say that the variables Y and Z are readily eliminated at the price of transforming each summand of the objective function (7.5) to a piecewise linear, convex function in X_{rs}; each function has two linear portions, and the linearity of the objective function is restored by expressing X_{rs} as a sum of two variables,

$$X_{rs} = X_{rs}(1) + X_{rs}(2) .$$

The problem then comes down to the following, in the variables $X_{rs}(1)$, $X_{rs}(2)$, and f: minimize

$$\lambda f - \sum_{r,s}\{M_{rs}X_{rs}(1) + L_{rs}X_{rs}(2)\} \tag{7.6}$$

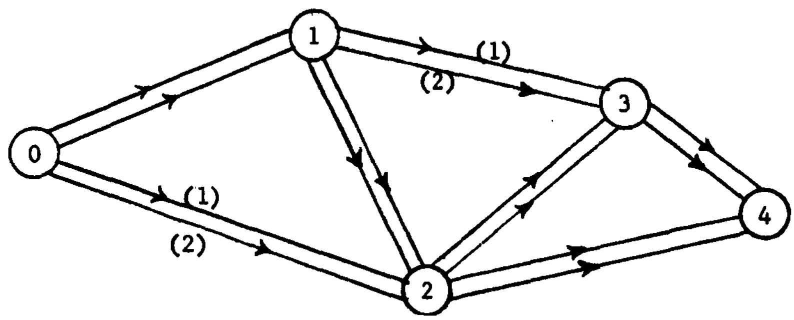
subject to

$$\sum_{s}\{X_{rs}(1) + X_{rs}(2)\} - \sum_{t}\{X_{tr}(1) + X_{tr}(2)\} = \begin{cases} 0, & r\neq 0,N, \\ -f, & r = N, \end{cases} \tag{7.7}$$

and

$$\begin{aligned} 0 &\leq X_{rs}(1) \leq C_{rs} , \\ 0 &\leq X_{rs}(2) . \end{aligned} \tag{7.8}$$

What is interesting now is that this last form of the problem can be recognized as one of network flow at minimum cost, the type of problem discussed in the previous section. The network involved is a modification of W obtained by imposing two similarly directed branches (1) and (2) in place of each individual branch of W; $X_{rs}(1)$ is the flow along branch (1) from r to s, and $X_{rs}(2)$ that along branch (2), and (7.7) expresses the balance of flow at the nodes for a total network flow of f; (7.8) are capacity restraints.



The primal-dual procedure and labeling technique for network flow described earlier can be adapted to solve this last problem parametrically in λ. The procedure yields the optimal occurrence times $t_r(\lambda)$ for the project network as well, with $t_N(\lambda) = \lambda$; these, in turn, determine the optimal project schedule by taking the duration times $T_{rs}(\lambda)$ as in (7.3). In fact, the dual variables used in the algorithm are the occurrence times t_r of the project events themselves; the complementary slackness conditions imposed on the primal variables $X_{rs}(1)$, $X_{rs}(2)$ in terms of these are

$$\left.\begin{array}{l} M_{rs} + t_r - t_s < 0 \text{ implies } X_{rs}(1) = 0 \\ L_{rs} + t_r - t_s < 0 \text{ implies } X_{rs}(2) = 0 \\ M_{rs} + t_r - t_s > 0 \text{ implies } X_{rs}(1) = C_{rs} . \end{array}\right\} \quad (7.9)$$

Thus, each iteration of the algorithm amounts to maximizing the flow f in (7.7) and (7.8) with the added restrictions implied by (7.9) for given values of the dual variables t_r.

The algorithm can be outlined in somewhat greater detail as follows. To start the algorithm, initial values of the dual variables t_r are taken in such a way to accommodate all the maximum values M_{rs} as activity durations, i.e., so that $T_{rs} = M_{rs}$ results from (7.3); this corresponds to beginning with the largest finish time $t_N = \lambda^*$, that need be considered. Next, the maximal flow problem (7.7), (7.8) is considered under the suppression restraints of (7.9). The maximal flow is found by a labeling technique (in this first round $X_{rs}(1) = X_{rs}(2) = 0$ are convenient initial flow values); when nonbreakthrough occurs in the labeling procedure, the maximum f has been reached and new values t_r' are defined by decreasing t_N and certain selected t_r by some positive amount. The new dual values define a new finish time $\lambda' = t_N'$, and the labeling computation is now repeated for the new maximal flow problem under the suppressions imposed by the new dual values. At the beginning of each round, the optimal flow values $X_{rs}(1)$, $X_{rs}(2)$ determined in the last round may be used as initial values for the new maximal flow problem.

The above procedure will continue until t_N reaches λ_o, the least value of λ permitting a feasible project schedule; this will be recognized by the fact that dual values with $t_N = \lambda_o$ will initiate a problem for which the maximal flow is ∞. At this point, the entire computation has achieved the original goal, the description of an optimal schedule over the full range of allowable finish times λ (for $\lambda \geq \lambda^*$, the schedule is the same as that for $\lambda = \lambda^*$). It may be remarked that the minimum project cost, as a function of λ, is completely specified by its values at the successive points t_N generated by the algorithm; the function

is linear between these points (however, the full linear portions of this function may actually range over longer stretches, with break points occurring only at certain ones of the successive t_N).

References

1. Ford, L. R. and Fulkerson, D. R., <u>Flows in Networks</u>, Princeton, N. J.: Princeton University Press, 1962.

2. Fulkerson, D. R., A Network Flow Computation for Project Cost Curves. <u>Management Science</u>, January 1961, <u>7</u>, 167-178.

3. Jewell, W. S., Optimal Flow Through Networks with Gains. <u>Operations Research</u>, July-August 1962, <u>10</u>, 476-499.

4. Kelley, J. E., Jr., Critical-Path Planning and Scheduling: Mathematical Basis. <u>Operations Research</u>, May-June 1961, <u>9</u>, 296-320.

UNCLASSIFIED

System Development Corporation,
Santa Monica, California
DUALITY AND NETWORK FLOW.
Scientific rept., TM-1042/201/00,
by W. Karush. 15 March 1963, 37p.
4 refs.

Unclassified report

DESCRIPTORS: Mathematical
Prediction. Linear Programming.

Describes the notions of duality
and network flow and shows how
these are used in the construction

UNCLASSIFIED

---

of algorithms for certain classes
of LP (linear programming) problems.
Reports that the problems include
the transportation problem, network
flow at minimum cost, and scheduling
of activities at minimum cost in a
PERT-like, or critical-path, system
of project organization.

UNCLASSIFIED

UNCLASSIFIED